REMARKS

It would appear that the Examiner may have misquoted the patent, and taken the quote out of context. Examiner's quote is "in the OLE DB, the JAVA applications were limited to accessing only relational data sources utilizing JBDC" (page 2 of Examiner's document, second paragraph). This misquote implies that JAVA applications use OLE DB. They do not and cannot, there is no JAVA-based API for OLE DB. *Sun Microsystems* maintains a list of commercial and publicly available JDBC drivers at http://developers.sun.com/product/jdbc/drivers. At the time the application was filed, only one vendor claimed to have a JDBC driver that worked with OLE DB, now there are two. Neither works with hierarchical data:

- Attunitty Connect (http://www.attunity.com/Products/AttunityConnect.Asp) provides access to SQL and XML data, neither of which is hierarchical.
- jadoZoom from infoZoom (http://www.infozoom.de/en_jadoZoom.shtml) also does not provide access to hierarchical data.

The actual statement from the patent application is as follows:

> JDBC (JAVA Database Connectivity) is the JAVA Application Program Interface (API) used to access data in relational databases. While typical Windows programs written in C++ and Visual Basic had, in the OLE DB (Online Linking and Embedding Database), a standardized mechanism for accessing data in a variety of data sources, however, the JAVA applications were limited to accessing only relational data sources. (page 2, lines 15-22).

There are three points in these sentences:

1. JDBC provides access to data in relational databases. Here is a definitive reference:

   - http://java.sun.com/products/jdbc/index.jsp, first paragraph reads:

     > JDBC technology is an API (included in both J2SE and J2EE releases) that provides cross-DBMS connectivity to a wide range of SQL databases and access to other tabular data sources, such as spreadsheets or flat files. With a JDBC technology-enabled driver, you can connect all corporate data even in a heterogeneous environment.

     Note that "tabular data" does not include "hierarchical data".

2. OLE DB enables C++ and Visual Basic application to access relational as well as hierarchical data sources.

   References:

   - http://msdn.microsoft.com/library/default.asp?url=/library/en-us/oledb/htm/dasdkoledboverview.asp, first paragraph reads:

     > OLE DB is a set of COM-based interfaces that expose data from a variety of sources. OLE DB interfaces provide applications with uniform access to data stored in diverse information sources, or data stores. These interfaces support the amount of DBMS functionality appropriate to the data store, enabling the data store to share its data.

- http://msdn.microsoft.com/library/default.asp?url=/library/en-us/oledb/htm/oledbprovthe_data_shaping_service_for_ole_db.asp

  The Data Shaping Service for OLE DB allows an application to create relationships that had not previously existed between keys, fields, or rowsets. It supports the construction of hierarchical rowset objects from your data provider.

This statement alludes to the hierarchical capabilities of OLE DB, in the case, availability of a tool to make tabular data appear as hierarchical data to the application using OLE DB:

- http://msdn.microsoft.com/library/default.asp?url=/library/en-us/oledb/htm/oledbprovpersisted_hierarchical_data.asp

- http://msdn.microsoft.com/library/default.asp?url=/library/en-us/oledb/htm/oledbhierarchical_rowsets.asp

These pages introduce the programming steps necessary to access hierarchical data.

3. Since Java applications use JDBC, via that API they are limited to accessing relational data. This statement reflects back on point 1, which is that JDBC is for accessing relational data. No effort went into JDBC to support non-relational data, such as hierarchical data.

It is hoped that the above information will help clarify Examiners questions about the status of the prior art.

Respectfully submitted,

By Alfred W. Kozak

Alfred W. Kozak
Reg. No. 24,265
(858) 451-4615
(949) 380-5822

Certificate of Mailing (37 CFR 1.8a)

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: MAIL STOP AMENDMENT, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Patti S. Preddy

Date: February 8, 2006 Patti S Preddy